REMARKS

In the Office Action, claim 56 was objected to for depending from a claim that had been withdrawn from consideration. With the present Amendment, claim 56 has been rewritten in independent form. As such, the objection to claim 56 has been overcome.

Claims 41-54 and 56 were rejected under 35 U.S.C. §102(e) as being anticipated by Brumme et al. (hereinafter Brumme, U.S. Patent No. 6,134,559).

CLAIMS 41-45

Independent claim 41 provides a method of instantiating and initializing a programming object. The method includes selecting an object data set for an object from a plurality of object data sets for the object's class. Each object data set includes the same unique identifier for the object's class, where only objects of the object's class can be instantiated using the unique identifier. The object is instantiated based on the unique identifier and then is initialized using at least one attribute in the selected object data set.

Brumme does not show or suggest the invention of claim 41. In particular, Brumme does not show a plurality of object data sets that have the same unique identifier for an object's class where only objects of the object's class can be instantiated using the unique identifier.

In the Office Action, the primary key identification in Table 4 of column 33 was cited as a unique identifier for an object's class. However, this primary key identification is not used to instantiate objects of the object's class and further is not limited to being used in instantiating objects of a particular object's class.

As indicated in column 5, lines 3-11 of Brumme, data source adapters execute client requests for data by browsing metadata to determine the class type specified by the client. An

instance of that class type is then created and then is populated with data from a data source. In columns 33 and 34 of Brumme, Table 4 is used to acquire the data that will populate an instance of an object. Brumme does not say that the map represented by Table 4 is used to instantiate the object itself. Instead, the map is only used to acquire data that is used to populate an instantiated object. As such, the primary key identification of Table 4 is not used to instantiate the object and as such can not be considered a unique identifier as found in claim 41 because claim 41 requires the instantiation of an object based on the unique identifier.

In addition, the primary key identification of Table 4 is not shown by Brumme to be limited to only one class type. The primary key identification identifies a table in a relational database. Brumme does not indicate that this table cannot be associated with other object types. For instance, a class type for a business unit may reference the employees of the business unit using the employees of each department in the business unit. Thus, the class type for the business unit may map to the same table as the class type for the departments. If that is the case, the same primary key identification would be used to identify the table for two different class types.

This is substantially different from claim 41 where only objects of the object's class can be instantiated using the unique identifier for the object's class.

Note that those skilled in the art would not limit Brumme such that the primary key identification for a relational table could only be used with a single class, since that would limit the data that could be associated with an object. In Brumme, the C-Dept. class provides a collection of employees for a given name of a department. This name is mapped to a unique department ID in the map of table 4. This unique department ID is presumably then used in another table to find the employees

assigned to that department ID. If the relational database only allows the department object class to access the department table, no other object class can obtain the department ID. This means that other information associated with a department, such as the serial numbers of the computers in the department, cannot be obtained using the same department ID. This would greatly reduce the number of objects that can be created and the ability of the objects to represent the entire relational database. As such, those skilled in the art would not limit access to the tables to a single class type. This means that the primary key identification for the relational table would not be forced to be used with only one object class.

In light of the fact that the primary key identification in Brumme is not used to instantiate an object and the fact that the primary key identification in Brumme is not limited to being used for only one object class, the invention of claim 41 is not shown or suggested by Brumme. As such, claim 41 and claims 42-46, which depend therefrom are patentable over Brumme.

Dependent claim 42 is additionally patentable over Brumme. In dependent claim 42, selecting an object data set comprises locating an object data set, instantiating an object token that is separate from the object, and initializing the object token to point to the object data set. Brumme does not show or suggest these steps. In the Office Action, it was asserted that Brumme shows a step of instantiating an object token when it converts an object from a data source to the uniform object model. However, this is different from the invention of claim 42 since in claim 42, the object token is separate from the object. Under claim 42, the object token is instantiated as a form of indirection that allows the attributes associated with an object to be queried without having to instantiate the object itself. Thus, an object data set can be located and selected

before the instantiation of the object itself. Note that under Brumme, clients cannot obtain information about an object's attributes until after the object has been instantiated. This is substantially different from claim 42 where an object token is instantiated that is separate from the object and is initialized to point to the object data set.

Claim 43 is also additionally patentable over Brumme. In the Office Action, no specific part of Brumme was cited as showing the limitations of claim 43. In claim 43, instantiating an object comprises calling an instantiation method exposed by an object token, where the object token has been initialized to point to an object data set. Brumme does not show or suggest this form of instantiating an object. As such, claim 43 is additionally patentable over Brumme.

CLAIMS 46 AND 48-51

Independent claim 46 provides a computer readable medium having a data structure. The data structure includes a first set of object data for an object class where the first set of object data comprises a first entry containing a unique identifier for the object class that must be known in order to instantiate an instance of the object. The first entry also contains at least one attribute of the object class. A second set of object data is also included in the data structure for the object class. The second set of object data includes a second entry containing the same unique identifier as the first entry and at least one attribute of the object class that is different from the at least one attribute of the first set of object data.

Brumme does not show or suggest the invention of claim 46. In particular, Brumme does not show two entries that contain a same unique identifier for an object class where that unique identifier must be known in order to instantiate an instance of the object.

In the Office Action, it was asserted that column 35, line 60 to column 36, line 8 showed the limitations of claim 46. Specifically, it was asserted that the same object is located in different locations in the cited section with the object being located in a foreign object system and a uniform object model system.

In the cited section, which constitutes claim 1 of Brumme, foreign objects from a foreign object system are received into an integrated object oriented system and converted into uniform object model objects. This conversion is more than a copying of the object. Instead, as shown in column 30, lines 10-16, in order to add a foreign object, a foreign object adapter creates a class type for the object-oriented environment by mapping the foreign object class type to a class type compatible with the object-orientated environment. Thus, the class type in the object-oriented environment is different than the class type in the foreign object system. The same object is not located in different locations, but instead, one object class in one object environment is mapped into a different object class in a different object environment. The mapping from one class type to another class type as found in Brumme is not the same as claim 46 in which two different sets of object data both contain the same unique identifier for an object class but also contain different attributes for the object class. As such, claim 46 is not shown or suggested in Brumme. Claims 48-51 which depend from claim 46 are also patentable over Brumme for the same reasons.

Dependent claim 48 is additionally patentable over Brumme. In claim 48, one of the attributes in the set of object data is a data file for initializing an object. Brumme does not show or suggest an object attribute that is a data file for initializing the object. As such, claim 48 is additionally patentable over Brumme.

CLAIMS 52 AND 54

Independent claim 52 is a computer readable medium having a computer loadable object token. The object token includes computer-executable instructions for setting the object token to point to a set of object data related to an object, for retrieving attributes from the set of object data, for instantiating the object based on a unique identifier in the set of object data, and for passing a pointer to the object token to the object.

Brumme does not show or suggest the invention of claim 52. In particular, Brumme does not show an object token that has instructions for instantiating an object based on a unique identifier in a set of object data and instructions for passing a pointer to the object token to the object. As such, claim 52 and claim 54 which depends therefrom are patentable over Brumme.

CLAIM 56

Independent claim 56 is directed to a computer readable medium having a computer loadable token enumerator. The token enumerator includes computer-executable instructions for locating object attributes for an object that are located outside of a static attribute storage location. Further, the enumerator has instructions for instantiating an object token, for initializing the object token to point to the object attributes, and for providing to an instance of the object the pointer to the instantiated object token as reference to the object attributes.

Brumme does not show or suggest a token enumerator that has the instructions shown in claim 56. In particular, Brumme does not show a token enumerator that can instantiate and initialize an object token to point to object attributes and that can provide a pointer to the instantiated object token to an object. As such, claim 56 is patentable over Brumme.

CONCLUSION

Based on the above remarks, claims 41-46, 48-52, 54 and 56 are patentable over the cited art. Reconsideration and allowance of the claims is respectfully requested.

The Director is authorized to charge any fee deficiency required by this paper or credit any overpayment to Deposit Account No. 23-1123.

Respectfully submitted,

WESTMAN, CHAMPLIN & KELLY, P.A.

By: _____

Theodore M. Magee, Reg. No. 39,758
Suite 1600 - International Centre
900 Second Avenue South
Minneapolis, Minnesota 55402-3319
Phone: (612) 334-3222 Fax: (612) 334-3312

TMM:sew